**SuperADS广告sdk，Android开发接入文档**

SuperADS广告sdk提供了横幅广告（Banner），插屏广告（Interstital），原生广告（Native Banner, Native Feed）等3种广告形式，通过集成广告sdk展示广告可以获得收益。本文描述了Android开发者如何集成sdk和展示广告。要看广告效果，可以在Google Play上面搜索“爱思店”APP。

编码示例：<https://github.com/156076769/superads_standalone_demo>

1. **集成前获取PublisherId和ApplicationId**

SuperADS为每个开发者分配一个Publisher Id, 为每个应用分配一个Application Id。在集成sdk前，请联系SuperADS运营获取此2个Id。同时每个开发者publisher有自己唯一的公钥来和SuperADS通信，请联系运营获取此key。

1. **依赖sdk和初始化**

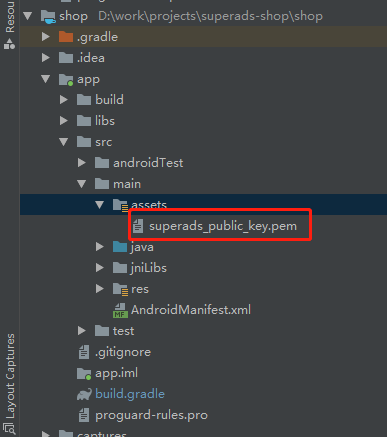
2.1在项目的build.gradle里面添加依赖

1. implementation 'com.superads.android:adsdk:0.1.2'
2. implementation 'com.github.bumptech.glide:glide:4.9.0'
3. implementation 'com.squareup.retrofit2:retrofit:2.5.0'
4. implementation 'com.squareup.retrofit2:converter-gson:2.5.0'

2.2混淆配置

1. -keep **class** com.bumptech.\*\* { \*; }
2. -keep **public** **class** \* **implements** com.bumptech.glide.module.GlideModule
3. -keep **public** **enum** com.bumptech.glide.load.resource.bitmap.ImageHeaderParser$\*\* {
4. \*\*[] $VALUES;
5. **public** \*;
6. }
7. -keep **class** okhttp3.\*\* { \*; }
8. -keep **class** retrofit2.\*\* { \*; }
9. -keep **class** com.superads.android.adsdk.ads.utils.CryptoUtils { \*; }
10. -keep **class** com.superads.android.adsdk.ads.providers.SuperAds { \*; }
11. -keep **class** com.superads.android.adsdk.ads.providers.models.AdRequest { \*; }
12. -keep **class** com.superads.android.adsdk.ads.providers.models.AdRequest$Builder { \*; }
13. -keep **class** com.superads.android.adsdk.ads.rendering.view.BannerAdView{ \*; }
14. -keep **class** com.superads.android.adsdk.ads.rendering.view.InterstitialAdLoader { \*; }
15. -keep **class** com.superads.android.adsdk.ads.providers.models.NativeAdRequest { \*; }
16. -keep class com.superads.android.adsdk.ads.rendering.view.VideoAdLoader { \*; }
17. -keep **class** com.superads.android.adsdk.ads.providers.models.NativeAdRequest$Builder { \*; }
18. -keep class com.superads.android.adsdk.ads.providers.models.NativeAdRequest$OnNativeAdLoadedListener { \*; }
19. -keep class com.superads.android.adsdk.ads.providers.models.NativeAd { \*; }
20. -keep **class** com.superads.android.adsdk.ads.rendering.view.NativeAdLoader { \*; }
21. -keep **class** com.superads.android.adsdk.ads.rendering.view.AdListener { \*; }
22. -keep **class** com.superads.android.adsdk.ads.providers.SuperAdsConfig { \*; }

2.3将下载的公钥文件重名为superads\_public\_key.pem，放在assets目录下，命名为：superads\_publick\_key.pem



2.4初始化

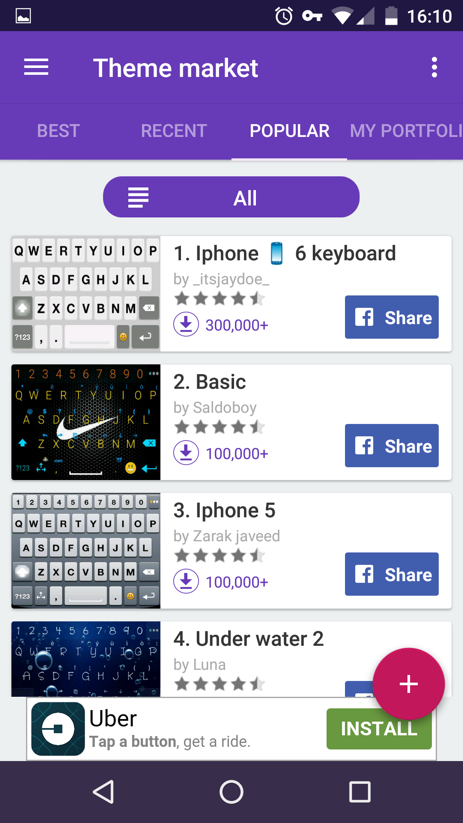
1. SuperAds.init(context,
2. "分配给你的publisherId", "分配给你的applicationId");

建议在application的onCreate里面调用此初始化

1. **请求和展示广告**

**3.1横幅广告**

横幅广告是一个320\*50（横屏下是768\*90）尺寸的广告，在屏幕的顶部或者底部展示。



第一步：在你要展示广告的页面布局中加入容纳广告的FrameLayout，并且将此FrameLayout放在顶部或者底部。

1. <com.superads.android.adsdk.ads.rendering.view.BannerAdView
2. android:id="@+id/card\_banner"
3. android:layout\_width="wrap\_content"
4. android:layout\_height="wrap\_content"
5. android:layout\_gravity="center" />

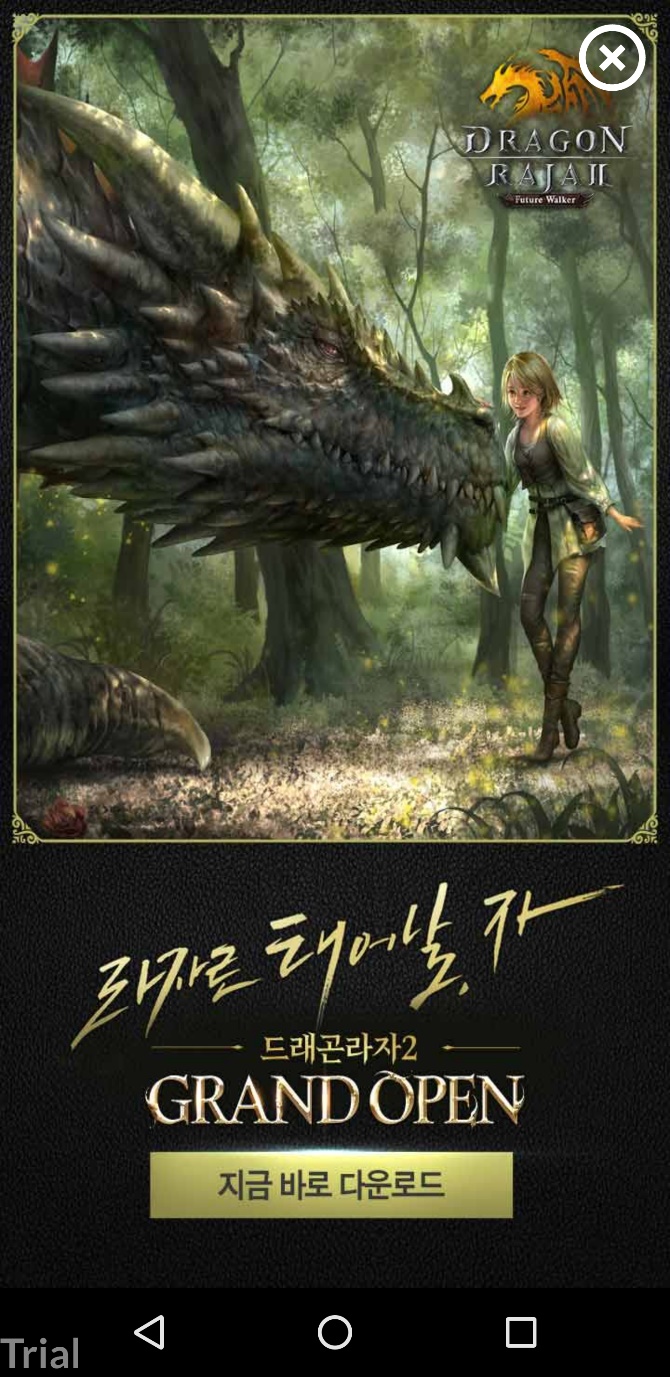
第二步：调用请求接口，在回调中展示view

1. Button btnBanner = findViewById(R.id.btn\_load);
2. btnBanner.setOnClickListener(view -**>** {
3. final BannerAdView bannerAdView = findViewById(R.id.card\_banner);
4. AdRequest.Builder builder = new AdRequest.Builder(SuperAds.genRandomPlacementId());
5. bannerAdView.loadAd(builder.build(), new AdListener() {
6. @Override
7. public void onAdLoaded() {
8. bannerAdView.setVisibility(View.VISIBLE);
9. }
11. @Override
12. public void onAdFailedToLoad(int errorCode) {
13. Log.e("NativeExamplesActivity", "error generating ad, error code=" + errorCode);
14. }
15. });
16. });

广告位id如果没有，可用SuperAds.genRandomPlacementId()生成随机id

**3.2插屏广告**

插屏广告是1024\*768（竖屏下）的广告，全屏显示。

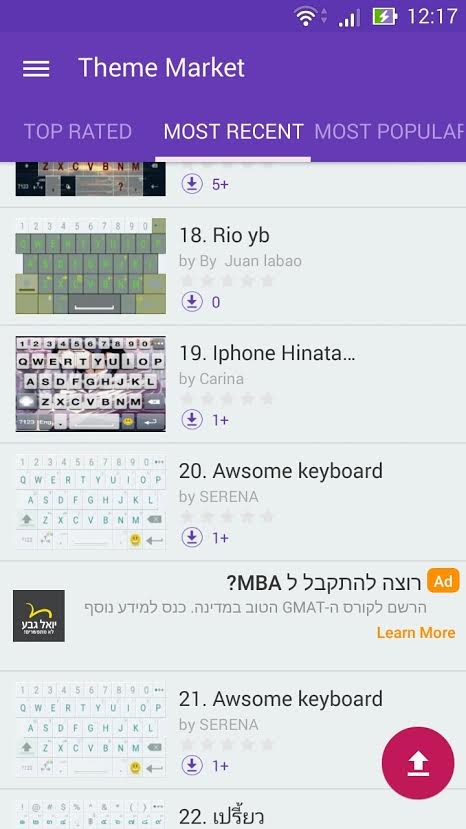


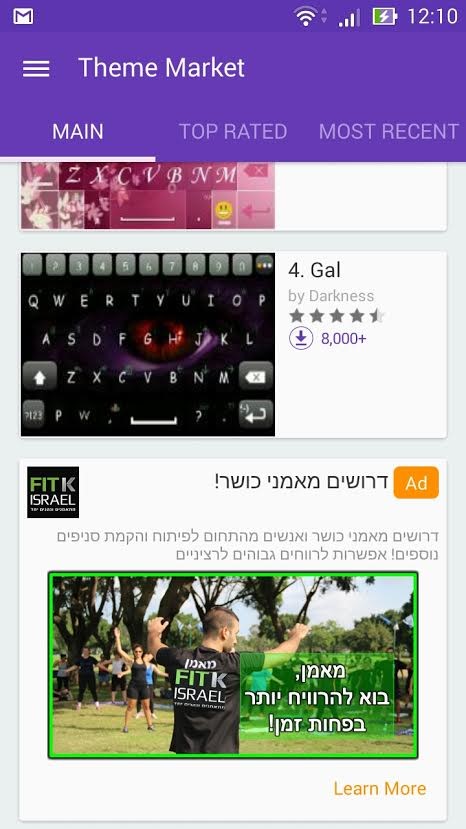
在跳转你的activity之前，调用如下。当用户关闭广告时，onAdClosed回调触发，此时你startActivity到你的目标activity

1. Button btnLoad = findViewById(R.id.btn\_load);
2. btnLoad.setText("Load Interstitial");
3. btnLoad.setOnClickListener(view -**>** {
4. AdRequest.Builder builder = new AdRequest.Builder(SuperAds.genRandomPlacementId());
5. this.interstitialAd = new InterstitialAdLoader(this);
6. interstitialAd.loadAd(builder.build(), new AdListener() {
7. @Override
8. public void onAdLoaded() {
9. Toast.makeText(InterstitialActivity.this, "interstitial Ad load successfully", Toast.LENGTH\_SHORT).show();
10. }
12. @Override
13. public void onAdFailedToLoad(int errorCode) {
14. Log.e("NativeExamplesActivity", "error generating ad, error code=" + errorCode);
15. }
17. @Override
18. public void onAdClosed() {
19. super.onAdClosed();
20. }
21. });
22. });
24. View btnShow = findViewById(R.id.btn\_show);
25. btnShow.setOnClickListener(view -**>** {
26. if (interstitialAd != null && interstitialAd.isloaded())
27. interstitialAd.show();
28. });

**3.3 原生广告**

原生广告是嵌入到你的list中的广告样式，也叫信息流广告，分为Native Banner和Native Feed两种，是目前较流行的方式。





服务器返回的原生广告Ad素材包含下面4个元素：

1. App Icon：应用小图标



2. APP Title：应用名称



3. APP Description：应用描述



1. APP Big Image：应用大图（Banner形式没有，Feed形式有）



开发者可以用这4（3）个原生来构建一个Recycle View Item，嵌入到自己原来的list中，选择合适的位置展示，并且适配成自己list item的风格使其显示自然。

下面是一个实例：

第一步：定义一个符合自己list item风格的item layout用来容纳原生广告素材的4元素

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<LinearLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"**>**
6. **<FrameLayout**
7. android:id="@+id/adContainer"
8. android:layout\_width="match\_parent"
9. android:layout\_height="wrap\_content"**/>**
11. **</LinearLayout>**
12. **<?xml** version="1.0" encoding="utf-8"**?>**
13. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
14. xmlns:tools="http://schemas.android.com/tools"
15. android:layout\_width="match\_parent"
16. android:layout\_height="wrap\_content"
17. android:layout\_gravity="center"
18. android:layout\_margin="0dp"
19. android:background="@color/white"
20. android:padding="0dp"**>**
22. **<TextView**
23. android:id="@+id/ad\_txt\_ad\_indicator"
24. android:layout\_width="wrap\_content"
25. android:layout\_height="wrap\_content"
26. android:layout\_alignParentTop="true"
27. android:layout\_alignParentEnd="true"
28. android:layout\_alignParentRight="true"
29. android:layout\_margin="1dp"
30. android:gravity="center"
31. android:text="Sponsored"
32. android:textColor="@color/blue"
33. android:textSize="12sp"
34. tools:ignore="HardcodedText" **/>**
36. **<ImageView**
37. android:id="@+id/ad\_img\_icon"
38. android:layout\_width="wrap\_content"
39. android:layout\_height="wrap\_content"
40. android:layout\_alignTop="@+id/ad\_txt\_title"
41. android:layout\_alignBottom="@+id/ad\_txt\_title"
42. android:layout\_margin="5dp"
43. android:scaleType="fitXY" **/>**
45. **<TextView**
46. android:id="@+id/ad\_txt\_title"
47. android:layout\_width="wrap\_content"
48. android:layout\_height="wrap\_content"
49. android:layout\_below="@+id/ad\_txt\_ad\_indicator"
50. android:layout\_toEndOf="@+id/ad\_img\_icon"
51. android:layout\_toRightOf="@+id/ad\_img\_icon"
52. android:padding="4dp"
53. android:textColor="#006064"
54. android:textSize="24sp"
55. tools:text="Awesome application that you have to try is now available for free download" **/>**
57. **<ImageView**
58. android:id="@+id/ad\_img"
59. android:layout\_width="match\_parent"
60. android:layout\_height="150dp"
61. android:layout\_below="@+id/ad\_txt\_title"
62. android:layout\_centerHorizontal="true"
63. android:layout\_margin="8dp"
64. android:scaleType="fitXY" **/>**
66. **<TextView**
67. android:id="@+id/ad\_txt\_description"
68. android:layout\_width="wrap\_content"
69. android:layout\_height="wrap\_content"
70. android:layout\_below="@+id/ad\_img"
71. android:layout\_margin="12dp"
72. android:ellipsize="end"
73. android:padding="2dp"
74. android:textColor="#009688"
75. android:textSize="16sp"
76. tools:text="Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat." **/>**
78. **<TextView**
79. android:id="@+id/ad\_txt\_cta"
80. android:layout\_width="wrap\_content"
81. android:layout\_height="wrap\_content"
82. android:layout\_below="@id/ad\_txt\_description"
83. android:layout\_alignParentEnd="true"
84. android:layout\_alignParentRight="true"
85. android:layout\_margin="8dp"
86. android:background="@drawable/cta\_square\_green"
87. android:gravity="center"
88. android:paddingStart="10dp"
89. android:paddingLeft="10dp"
90. android:paddingTop="5dp"
91. android:paddingEnd="10dp"
92. android:paddingRight="10dp"
93. android:paddingBottom="5dp"
94. android:textAllCaps="true"
95. android:textColor="@color/white"
96. android:textSize="24sp"
97. android:textStyle="bold"
98. tools:text="Download now" **/>**
100. **</RelativeLayout>**

第二步：创建广告的ViewHolder并且请求服务器返回元素填充它

1. private void loadNativeAds() {
2. NativeAdRequest.Builder builder = new NativeAdRequest.Builder(SuperAds.genRandomPlacementId(), new NativeAdRequest.OnNativeAdLoadedListener() {
3. @Override
4. public void onNativeAdLoaded(NativeAd nativeAd) {
5. refreshRecycleView(nativeAd);
6. }
7. });
9. final NativeAdLoader nativeAd = new NativeAdLoader(NativeActivity.this);
10. nativeAd.loadAd(builder.build(), new AdListener() {
11. @Override
12. public void onAdLoaded() {
13. Log.i("NativeAdapter", "native ad loaded");
14. }
16. @Override
17. public void onAdFailedToLoad(int errorCode) {
18. Log.e("NativeAdapter", "error generating ad, error code=" + errorCode);
19. }
20. });
21. }
23. private void refreshRecycleView(NativeAd nativeAd) {
24. DataType dataType = DataType.NATIVE\_BANNER;
25. if (getIntent().hasExtra("adType")) {
26. dataType = (DataType) getIntent().getSerializableExtra("adType");
27. }
29. List**<ItemData>** items = new ArrayList**<>**();
30. for (int i = 0; i **<** **10**; i++) {
31. ItemData data = new ItemData();
32. data.type = 1;
33. data.data = "txt";
34. items.add(data);
35. }
36. ItemData adItem = new ItemData();
37. adItem.type = 100;
38. adItem.data = nativeAd;
39. items.add(3, adItem);
41. switch (dataType) {
42. case NATIVE\_BANNER:
43. reycler.setAdapter(new NativeAdapter(items));
44. break;
45. case NATIVE\_FEED:
46. default:
47. reycler.setAdapter(new NativeAdapter(items));
48. break;
49. }
50. }
51. @Override
52. public RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
53. View v;
54. if (viewType == NORMAL) {
55. v = LayoutInflater.from(parent.getContext())
56. .inflate(R.layout.list\_item, parent, false);
57. return new ItemViewHolder(v);
58. } else {
59. final View adContainer = LayoutInflater.from(parent.getContext()).inflate(R.layout.superads\_native\_ad\_container, null);
60. final View adView = LayoutInflater.from(parent.getContext()).inflate(R.layout.superads\_native\_ad, null);
61. return new NativeAdViewHolder(adContainer, adView);
62. }
63. }
65. @Override
66. public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
67. ItemData data = list.get(position);
68. if (data.type != 100) {
69. String txt = (String) data.data;
70. ((ItemViewHolder) holder).bindData(txt);
71. } else {
72. NativeAd nativeAd = (NativeAd)data.data;
73. NativeAdViewHolder nativeAdViewHolder = (NativeAdViewHolder)holder;
74. nativeAdViewHolder.bindData(nativeAd);
75. }
76. }
77. public class ItemViewHolder extends RecyclerView.ViewHolder {
79. private TextView txtView;
81. public ItemViewHolder(@NonNull View v) {
82. super(v);
83. txtView = v.findViewById(R.id.tvTitle);
84. }
86. public void bindData(String txt) {
87. txtView.setText(txt);
88. }
89. }
91. public class NativeAdViewHolder extends RecyclerView.ViewHolder {
92. private ViewGroup adContainer;
93. private ViewGroup adView;
95. private TextView titleTextView;
96. private TextView descriptionsTextView;
97. private TextView callToActionTextView;
98. private ImageView iconImageView;
99. private ImageView bigImageImageView;
101. public NativeAdViewHolder(@NonNull View itemView, View adView) {
102. super(itemView);
104. this.adContainer = itemView.findViewById(R.id.adContainer);
105. this.adView = (ViewGroup)adView;
106. this.adContainer.addView(this.adView);
108. this.titleTextView = itemView.findViewById(R.id.ad\_txt\_title);
109. this.descriptionsTextView = itemView.findViewById(R.id.ad\_txt\_description);
110. this.callToActionTextView = itemView.findViewById(R.id.ad\_txt\_cta);
111. this.iconImageView = itemView.findViewById(R.id.ad\_img\_icon);
112. this.bigImageImageView = itemView.findViewById(R.id.ad\_img);
113. }
115. public void bindData(NativeAd nativeAd) {
116. nativeAd.renderTitle(titleTextView);
117. nativeAd.renderDesc(descriptionsTextView);
118. nativeAd.renderCta(callToActionTextView);
119. nativeAd.renderIcon(iconImageView);
120. nativeAd.renderBigImgUrl(bigImageImageView);
121. nativeAd.registerViewForInteraction(adView);
122. }
123. }

**3.4 视频广告**

视频广告会以全屏播放的形式展示给用户，分为横屏和竖屏，短视频（15秒以内）和长视频（大于15秒）。适合在页面转场的适合使用。

1. Button btnLoad = findViewById(R.id.btn\_load);
2. btnLoad.setText("Load Video");
3. btnLoad.setOnClickListener(view -**>** {
4. AdRequest.Builder builder = new AdRequest.Builder(SuperAds.genRandomPlacementId());
5. this.videoAdLoader = new VideoAdLoader(this);
6. videoAdLoader.loadAd(builder.build(), new AdListener() {
7. @Override
8. public void onAdLoaded() {
9. Toast.makeText(VideoActivity.this, "video Ad load successfully", Toast.LENGTH\_SHORT).show();
10. }
12. @Override
13. public void onAdFailedToLoad(int errorCode) {
14. Log.e("NativeExamplesActivity", "error generating ad, error code=" + errorCode);
15. }
17. @Override
18. public void onAdClosed() {
19. super.onAdClosed();
20. }
21. });
22. });
24. View btnShow = findViewById(R.id.btn\_show);
25. btnShow.setOnClickListener(view -**>** {
26. if (videoAdLoader != null && videoAdLoader.isloaded())
27. videoAdLoader.show();
28. });